

Real-Life Challenges in Automotive Release Planning

Kristina Marner
Dr. Ing. h.c. F. Porsche AG,
Porschestraße 911,
71287 Weissach, Germany
Email: kristina.marner@porsche.de

Sven Theobald
Fraunhofer IESE,
Fraunhofer-Platz 1,
67663 Kaiserslautern, Germany
Email:
sven.theobald@iese.fraunhofer.de

Stefan Wagner
University of Stuttgart,
Universitätsstraße 38,
70569 Stuttgart, Germany
Email: stefan.wagner@iste.uni-
stuttgart.de

□ **Abstract—Context:** The use of agile software development is increasing, even in regulated domains like the automotive domain. At the same time, traditional sequential processes are still in use. Collaboration between agile and hybrid projects within these complex traditional product development processes is difficult. Especially the creation and synchronization of a qualification phase plan is challenging. **Objective:** The aim of this study is to provide insights into the state of the practice to understand challenges related to the combined use of agile and traditional paradigms in release planning in the automotive domain. **Method:** Based on semi-structured interviews, an online survey with 39 respondents was conducted at Dr. Ing. h. c. F. Porsche AG. **Results:** We present the challenges identified in release planning, such as lack of transparency regarding the status quo of related projects. Furthermore, we motivate how agile development methods could improve collaboration between projects in release planning. **Conclusions:** There are many challenges in the context of co-existing agile and traditional projects. We discuss how agile practices like daily standup or continuous integration could address the identified challenges.

I. INTRODUCTION

TODAY, it is a competitive advantage to develop and put products on the market as early as possible. Agile software development methods and practices are commonly used to achieve this goal [1]. Practitioners want to benefit from increased project visibility, faster response to change, and shorter time to market [1] by adopting agile development practices.

Nonetheless, traditional approaches like the waterfall or the V-model are still predominant in highly regulated domains. Within these domains, the adoption of agile practices is hard to achieve and even not always desired [2]. To overcome the factors that hinder an agile transformation, regulated domains prefer adopting single agile practices [3] into their development processes [4]. This inevitably leads to a mixture of different development processes ranging from completely traditional processes to agile adaptations [5][6], which in turn results in more and more complex interfaces

[7] between all involved methodologies. The mixture of traditional and agile development practices is called hybrid development approaches [5]. Such approaches are commonly used in the automotive domain [8].

In the automotive domain, the complexity of software and systems is constantly increasing [9]. As automotive projects are generally large projects with many subprojects and suppliers, it is necessary to preserve the benefits of the existing rich development processes [10] to coordinate all involved parties. In addition, current software development in the automotive domain is intended to address safety-critical functionality by means of standardized processes to satisfy requirements given by the law.

Thus, it is a challenge to speed up software release cycles [2]. Creating and updating a common release plan that considers all dependencies is challenging, even more so when multiple parties work with different processes.

The aim of this work is to investigate the challenges in the release planning of automotive projects when traditional and agile processes co-exist.

The contribution of our work is as follows: We identify and analyze challenges in the qualification phase to identify improvements in the context of co-existing agile and traditional projects from the perspective of an automotive Original Equipment Manufacturer (OEM).

The remainder of this paper is structured as follows: Related work is presented in Section 2. Section 3 defines the research approach including the research questions and design, the research site and the participants, the data collection and analysis procedure, as well as the data collection instrument. The survey results are reported in Section 4. We conclude our work and outline future research in Section 5.

II. BACKGROUND AND RELATED WORK

In the automotive domain, a hybrid project environment consists of two conflicting parts. There is the strategic framework on one side consisting of processes with many milestones planned a long time in advance before projects related to production and distribution go live. This strategic framework represents the time and content requirements, such as the product development process and thus defines a

□ This work was not supported by any organization

superordinate process. On the other hand, there is the operational level, where projects are performed in the way that best fits the project's character. On this level, projects are developed in an agile, hybrid or traditional way. A solution has to be found that synchronizes both levels and which enables coordinated release planning.

The automotive domain is a strongly regulated domain. Therefore, this combination cannot start in a green field, as strategic frameworks define different phases of the development process.

The Qualification Phase (QP) is the repetitive integration and testing process of an Electronic Control Unit (ECU) network, its sensors and actuators.

This phase is typically defined at the beginning of a project. The maturity level is determined to release the ECU network for further testing, usage, and development. The maturity levels provide information about the development progress of functions and ECUs in relation to the target state.

The Additional Qualification Phase (AQP) is an extra qualification phase with a reduced testing scope if the level of maturity is found to be insufficient and refers to a reduced scope of ECUs. The reduced test scope refers to the inadequate target state and is defined application-specifically. An AQP is not planned in advance but established depending on the quality level of the QP. In such cases, it is necessary and has to be executed. The selection of the test cases and the duration of the tests strongly depend on the errors identified during the QP.

Release planning in a hybrid project environment has barely been considered to date in the literature. Software release planning matches features to releases under the condition that different types of constraints are considered [11]. Heikkilä et al. [12] identify "an obvious gap in the research of release planning in large-scale agile software development organizations" in a literature review. However, they did not consider the combination of releases consisting of software and hardware.

Sax et al. [13] describe software release and configuration management in the automotive domain. Bestfleisch, Herbst and Reichert [14] define requirements for controlling and monitoring dependencies on other release processes with the help of workflow support. Müller et al. [15] define requirements for IT support to improve release management in the automotive domain. Lindgren et al. [16] identified key aspects of release planning in the context of software and system development projects. Furthermore, they captured the state of the practice for release planning in industry.

There is literature dealing with release planning in agile software development projects, both for single projects and for scaled projects. Danesh et al. [17] evaluated the methods used by companies to plan new software releases. Heikkilä et al. [18] present a case study where the agile release planning process in a scaled Scrum environment was evaluated. Heikkilä et al. [19] describe the qualification phase and present a case study of multi-team agile release planning with the help of this practice.

Karvonen et al. [20] conducted a systematic literature study to identify agile release engineering practices. Ameller

et al. [21] conducted a literature study to report on software release planning models. Overall, there is no direct related work that considers release planning in co-existing traditional and agile processes in the automotive domain. Some work deals with agile release planning, but none of the identified sources deals with the targeted hybrid project environment.

The HELENA study [5][6][8] investigates the combined use of agile and traditional practices in hybrid processes, but does not consider the co-existence of agile and hybrid projects and their synchronization. Theobald and Diebold [7] investigate and classify problems at the interface of agile development and a traditional environment. The work of this paper can be classified in the problem field "project planning" at the interface "project team" [7].

The focus of the majority of publications on release planning models are various kinds of mathematical models and simulations [22], which are ineffectual in complex industries [23]. Practitioners reported that these approaches are either too simple to generate a benefit or so difficult that they cannot reconstruct the whole process created [24][25].

There is a research gap considering hybrid project environments where projects with different development paradigms meet. Our paper aims to address this research gap.

III. RESEARCH APPROACH

A. RESEARCH QUESTIONS

This paper aims to answer the following research question: *What are the challenges and consequences of the qualification phase in an automotive hybrid project environment?* To answer this question, three research questions were defined:

- RQ1. What are challenges concerning the qualification phase in a hybrid project environment?
- RQ2. What are the specific challenges of agile projects embedded in a traditional development context?
- RQ3. How could agility address the identified challenges?

B. RESEARCH DESIGN

To answer the research questions, we selected a two-step research approach. First, we set up an exploratory, qualitative interview study within a German automotive OEM. An interview guide for identifying challenges and problems with regard to the release planning process was specified. The interview guide was tested in a pilot interview. Emerging issues, such as vague phrases, were addressed before the qualitative interview study was conducted. In the second step, an online survey questionnaire was developed to validate the challenges identified from the qualitative interview study in detail.

The data collection instrument was a questionnaire containing 31 questions. The survey questionnaire contained open and closed questions structured into six categories (cf. **Table1**).

TABLE 1. SURVEY QUESTIONNAIRE

Category	ID	Question
Context	1	What is your current role? [free text]
	2	How long have you been working in that role? [free text]
	3	What are you working on in your project? [E/E ECU, software component, function, connect service, vehicle project]
	4	Please select a sector to classify your project. [powertrain electronics, body electronics, infotainment, project is safety-critical, others]
	5	What kind of development method do you use? (agile, hybrid, or traditional) [use of adapted agile methods, hybrid methods, traditional approaches]
	6	If you are using agile or hybrid methods, please specify the method. [free text]
Qualification Phase	7	What do you think about the current number of qualification phases (incl. additional qualification phase)? [too high, adequate too low]
	8	How often are you able to generate current software versions ready to deliver? [never, seldom, often, always]
	9	Do you receive feedback about the qualification phase on time? [never, seldom, often, always]
	10	How often should a qualification phase take place in order for you to be ready to deliver? [every week, once a month, every 3 months, at larger intervals]
	11	Would additional releases in terms of partial composites with reduced test scope be helpful for safeguarding dependent ECUs? [yes, partially, no]
Planning	12	Is an initial planning of content possible? [never, seldom, often, always]
	13	Does an initial planning of content make sense? [never, seldom, often, always]
	14	How often is the content of the initial planning still up-to-date at the beginning of a qualification phase? [never, seldom, often, always]
	15	How difficult is it to get planning information for the relevant counterparts? [very difficult, difficult, easy, very easy]
	16	To what extent do management decisions, external influencing factors, or externally determined decisions influence your development process? [no impact, weak impact, strong impact, very strong impact]
Integration	17	To what extent does bug fixing affect the timely implementation of planned functionalities for the next qualification phase? [no impact, weak impact, strong impact, very strong impact]
	18	It is inevitable that software versions are released that are suboptimal concerning quality or content. [yes, partially, no]
	19	What kind of activities dominate your daily routine during a qualification phase? [free text]
	20	Rate the following statement: Additional qualification phases are necessary. [yes, partially, no]
	21	Rate the following statement: Additional qualification phases are reasonable. [yes, partially, no]
Coordination	22	Is the status of development transparent to you at any time? [yes, partially, no]
	23	Is the status of development of your stakeholders transparent to you at any time? [yes, partially, no]
	24	How important is the transparency of the development status of your relevant counterparts to you? [totally unimportant, unimportant, important, very important]
	25	Rate the following statements: - Stakeholder/Interfaces are known [Disagree, rather disagree, rather agree, agree] - Quality of coordination is good. [Disagree, rather disagree, rather agree, agree]
Testing	26	Development can no longer handle the high number of bug reports. [Disagree, rather disagree, rather agree, agree]
	27	Problem resolution management can no longer handle the high number of bug reports. [Disagree, rather disagree, rather agree, agree]
	28	What are the reasons for the high number of tickets? [free text]
	29	Do all planned changes to the ECU network have to be fully tested for each qualification phase? [yes, partially, no]
	30	Do all types of tests have to be performed for every ECU for each qualification phase? [yes, partially, no]
	31	When do all ECUs have to be fully tested? [every qualification phase, depending on the changes, not mandatory]

The categories and questions were derived from the insights gained in the previous interviews. The questions were originally written in German. The questionnaire went through four review cycles by an independent researcher as well as by a specialist from the case company. Review comments were discussed by the authors and addressed to improve the questionnaire.

In the first category, we elicited the “Context”, such as role and experience of the participant, as well as project type, area, and the development method used (traditional vs. agile). The second category, “Qualification Phase”, aimed at evaluating how many qualification phases are feasible. The third category, “Planning”, was for evaluating the need to have an initial plan as well as external influences on such a plan. At a certain point in the development process, an initial planning of the functional scope of an ECU must be submitted for each release. In addition to general ECU information, deviations from the required functional, network and diagnostic maturity levels must also be specified. We examined the need for AQPs in the fourth category “Integration”. Integration is an upstream part of the actual process and represents the integration of one or more ECUs into a whole network. Transparency of the status quo and the quality of coordination were the focus of the fifth category, “Coordination”. Finally, we covered all questions related to “Testing” in the last category, trying to evaluate which kind and intensity of tests are necessary and if and why there are so many bug reports. The test phase focuses on the execution of the qualification phase and is therefore a main activity.

C. DATA COLLECTION AND ANALYSIS PROCEDURE

To identify the main challenges, the first researcher conducted 26 semi-structured interviews, which took between 30 and 60 minutes each. The information from each interview was incorporated into later interviews. Because these interviews did not allow for quantitative results, an online survey was conducted to confirm the challenges and to draw a more complete picture by consulting different participants. This allows for quantitative results, but gave every participant the chance to provide further qualitative results by sharing their experiences.

95 potential participants were selected based on their roles, to cover all perspectives. Then the participants were invited via an email motivating the goal of the study and outlining the contents and the time expected to answer the questionnaire. A reminder email was sent after one week. Also, one of the participants forwarded the questionnaire to an additional group of 25 people. The survey was open from November to December 2018.

After extracting the data from the online survey tool¹ into an Excel document, we analyzed the answers for completeness. There were 39 complete responses, meaning all six pages of the questionnaire had been answered and thus the survey had been officially finished. In addition,

there were 16 incomplete answers where the questionnaire was not finished. Of these 16 incomplete answers, 1 participant stopped after category 3 (Planning), another one stopped after category 5 (Integration), and all others had discontinued the questionnaire even earlier. Although we had access to the incomplete data sets, we decided to only consider the complete data sets for further analysis. Since the survey was distributed to 120 people with 39 respondents, our response rate was about 33%. Afterwards, we conducted a descriptive analysis of the individual questions and analyzed the textual answers to identify common opinions.

D. RESEARCH SITE AND PARTICIPANTS

This study was conducted at Dr. Ing. h. c. F. Porsche AG, a manufacturer that builds sports cars for everyday driving. The division EE within Dr. Ing. h. c. F. Porsche AG in Weissach, Germany, is responsible for the development process of electronic systems and its integration into the development process of the complete vehicle. For achieving this goal, transparent development, processes and hence accurate release planning are essential.

The target population of our survey included all roles involved in the qualification phase process of automotive products where the subprojects differed in terms of the development approaches used, including agile as well as traditional methods. The sample selected consisted of stakeholders from Dr. Ing. h. c. F. Porsche AG involved in release planning activities. The participants were expected to be motivated enough to answer the comprehensive questionnaire because they anticipated improvements based on the findings that reflect their current situation.

E. THREATS TO VALIDITY

As the results only represent one specific case, it might not be possible to generalize them. However, the fact that the case company has the same framework conditions (regulated domains, complex supplier relationships and high safety requirements) as similar OEMs, others could benefit from the findings. The issues that were identified in the earlier interviews were addressed in the questionnaire, whereas new survey participants did not have a chance to add more individual problems during the online survey. There might be a bias concerning the stakeholders who participated. Some roles are overrepresented, while other relevant roles were not represented by many participants. This might have led to results that are skewed towards the opinion of certain roles. Nonetheless, many different roles participated in the study, providing answers from many perspectives. As in all surveys, non-response bias could have led to missing the opinions of certain participants.

IV. SURVEY RESULTS

This section contains the demographics and context of the respondents, followed by the presentation and discussion of the results of this work structured along the research questions.

¹www.limesurvey.org

A. CONTEXT

The respondents' professional experience in their current role (Q1) was slightly below six years on average, with a minimum of one year and a maximum of 16 years (Q2). Most of the respondents had management roles (n=17; 44%), others were responsible for projects, products, functions, integration, testing, quality, data, processes, or other related disciplines. 10 participants (26%) represented the operational level. The remaining 12 respondents (30%) had roles with responsibilities related to the environment of qualification phases.

The respondents described their working environment using one or more categories (Q3). Most participants reported working in vehicle projects (n=24), development of E/E components (n=18), development of functions (n=14), development of software components (n=12), and connected services (n=8). Others (n=7) dealt with IT backend, cross-project integration, distributed functions, or quality.

14% of the respondents answered that their project was safety-critical. Most participants assigned their project to the area of infotainment (n=13), followed by electronics for car bodies (n=11) and electronics for engines (n=7). Regarding the 24 additional classifications, ten participants reported working on crosscutting topics (Q4).

Most respondents reported using traditional development or project management approaches such as the V-model or sequential approaches (n=26). Only six respondents used adapted agile methods, and seven persons used hybrid approaches, which was defined as strongly adapted agile methods or use of only single agile practices (Q5). This showed that only one third of the study participants were using agile concepts at the time.

Agile implementations were based on Scrum or the Porsche-specific adaption of agile methods. One person even reported scaled agile and lean at the unit level combined with an adapted Scaled Agile Framework (SAFe). Single agile practices like daily standups, user stories, backlogs, retrospectives, or the Scrum Master role were used in traditional projects. Some respondents reported using both agile and traditional approaches at different project levels. One answer stated that agile was being used at the team level together with the V-model for whole projects, while another respondent reported using a sprint-like approach within the V-model due to highly dynamic changes in requirements. Another respondent indicated the use of different development paradigms in different life cycle phases (Q6).

B. RQ1: CURRENT CHALLENGES

In the following, the current challenges will be presented and discussed along the categories of the survey questionnaire. RQ1. What are challenges concerning the qualification phase in a hybrid project environment?

1) QUALIFICATION PHASE

The majority of the participants (n=22; 56%) stated that the current number of releases (p.a.), including all additional

qualification phases and special qualification, is too high (Q7). On closer inspection, there is a discrepancy between the answers by managers and those by developers with responsibility for products or functions. The former (n=17) reported that the existing number of releases is too high (56%), while the latter said it is too low (25%).

An analysis of the comments field of this question shows results relating to the regulated defined number of releases. The developers confirmed their opinion and asked for a higher number of qualification phases. The management group agreed with the regulated defined numbers.

Further information concerning the ordinary number of qualification phases was given by the group of developers using agile methods. For the majority of those participants, the absolute number of qualification phases is too low to use agile methods properly.

The next issue concerned the delivery results in the required form (Q8). 60% of the survey participants answered that the required deliverable is seldom available in the required quality. In contrast, 40% replied that it is always or at least most of the time possible to create a delivery version for every requested release.

74% of the participants answered that they mostly receive feedback about qualification phases on time (Q9). The next question dealt with the number of qualification phases with regard to generate software version (Q10). Two-thirds of the participants stated that qualification phases should take place at least each quarter of the year. In contrast to the last question (Q11) in this category, 46% called for additional qualification phases with reduced test scopes.

2) PLANNING

This category highlights the characteristics around planning. The first question (Q12) aimed at evaluating the feasibility of initial planning at the beginning of the project. 50% of the participants in our study reported that initial planning is possible, and the other half answered that such a plan is rarely possible. At the beginning of a project, the decisions for or against a supplier have sometimes not been made yet. That is one reason why it is difficult to generate an initial planning. Another person replied that requirements for functions are the results of testing, which is done further on in the development process.

In a further question, the participants were asked if such initial planning would be meaningful (Q13). A significant majority (74%) stated that planning at the beginning of a project is reasonable because it is a resilient starting point for further steps. Participants also mentioned the existing change management process, which permits updates at any time.

The next question (Q14) regarding this topic dealt with the projected content before the next release in terms of timeliness. The results show that scheduled content is frequently impossible to implement in practice (80%). The majority of the participants stated that awareness still exists for high quality in planning. Planning updates have to pass a committee, which is one reason why change requests are not implemented in the current release. Also some areas,

“connected car”, are very dynamic, which is another reason for the bad current state of planning, which is not up-to-date. Receiving information about planning details from the relevant stakeholders is perceived as challenging (Q15). 74% of the respondents replied that obtaining information on time is difficult because there are no regulated tasks nor a consistent workflow for changing the relevant information.

Another issue is the impact of management decisions during the development cycle (Q16), which implies that these cannot be implemented easily. 90% of the respondents rated this influence as strong or very strong and reported that the development of new functionality suffers from having to deal with unexpected changes demanded by management. Some respondents complained about management decisions that change the backlog priority and have severe effects on further procedures.

3) INTEGRATION

This category contains the results relating to the challenges of software and hardware integration during a development cycle.

During a qualification phase, new software versions are tested at different levels of integration. The test results and even bug fixing have a great impact on the subsequent procedure (Q17). 87% of the participants answered that bug fixing affects their timely implementation related to the next release. Because there is no hold available in the project plan, this even leads to delays of the next scheduled functions (Q18).

Another question in this category dealt with the activities during a qualification phase (Q19). The main activities or tasks linked to the respective role are: Management is engaged in coordination and ensuring the scheduled scope with regard to the next release. At the operational level, tracking of test results and analysis of upcoming bug tickets are the main concerns. Both groups have to handle the subsequent deliveries.

Almost all interviewees (96%) admitted that delivering software versions with high quality is infeasible when they also have to provide the content planned for the next release. The results considered for integration have low maturity, due to the increasing pressure of costs and deadlines.

For this reason, additional qualification phases have been established subsequent to the original deadline. We wanted to know if such additional qualification phases are necessary (Q20) and reasonable (Q21). 65% of the participants considered additional qualification phases necessary and 35% were convinced that they are reasonable.

The main reasons given by the participants for subsequent integration were poor software quality, lack of adherence to delivery dates on the part of the suppliers, poor scheduling without buffering, and no complete bug fixing from the previous qualification phase.

4) COORDINATION

Transparency and coordination were the relevant aspects in this category (Q22). We asked whether the current

development status of the respondents' own team or dependent teams is sufficiently transparent. Only 26% (n=9) reported that their own development is transparent. The majority of the respondents rated transparency as only partial (n=20; 51%) or non-existent (n=10; 26%).

Next, the results of questions Q23 and Q24 are presented. The questions dealt with the transparency of the status of projects by relevant stakeholders and relevant counterparts. Here, only 15% (n=6) of the respondents answered that the development status of other projects is transparent for them. Most participants (n=19; 49%) reported partial transparency, while 36% (n=14) reported a lack of transparency. Reasons for the lack of transparency were missing time and coordination mechanisms, and the use of outdated content of the release plans.

The transparency of the status quo of a certain development project is very important and closely linked to the quality of a release. 95% of the respondents supported the statement that having a transparent software version at any time is important. It is necessary due to the complexity, dependency, and connectivity of software engineering.

Another question aimed at getting information about the communication structures within the company and involved persons from the release planning process (Q25). The participants had to rate whether they knew their interfaces and relevant stakeholders and whether the quality of the coordination was good. This rating had to be done for several interfaces: within the team, between team and testing, within the case company, within the company group, as well as towards external suppliers.

The results presented in Fig. 1 (bottom figure) demonstrate that communication quality decreases with longer communication paths: Communication within a project was perceived as good, but the quality was perceived as decreasing in communication within the company and even worse in communication with suppliers (internal means company group and external suppliers). Similarly, the relevant stakeholders and interfaces of the wider project context were reported less known than those within the team (see Fig. 1, top figure).

5) TESTING

This category assesses the testing situation. The first question aimed to evaluate whether the number of bug reports is still controllable by development (Q26) or problem resolution management (Q27). Overall, 56% (n=22) of the participants agreed (n=7) or rather agreed (n=15) that development is able to control the high number of bug reports. The remaining respondents had a tendency to disagree (n=9) or disagreed (n=8).

Concerning problem resolution management, most participants (n=25) disagreed (n=8; 21%) or had a tendency to disagree (n=17; 44%). The minority of the participants agreed (n=5; 12%) or rather agreed (n=9; 23%)

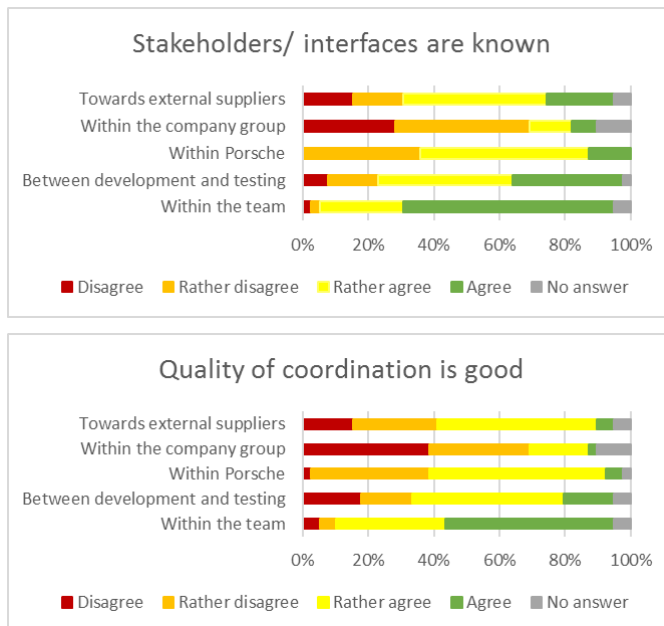


Fig. 1. Known interfaces (top) and quality of coordination (bottom)

Furthermore, the participants were asked about reasons for the high number of bug reports / tickets (Q28). The survey revealed that identifying errors is usually not done before the upcoming release due to insufficient development time, cost, and deadline pressure. It was reported that the intensity of testing by the supplier was not adequate. Other reasons given for the high number of error tickets are the rising complexity of the product itself, the lack of coordination within the team, and inadequate requirements engineering. Generally, it can be stated that the software quality before a qualification phase is insufficient and questionable, endangering the success of the qualification phase.

Software changes may have severe effects on interfaces, which is why tests have to be done. The need for testing the software changes to the full extent for every qualification phase (Q29) was not seen by 18% of the respondents (n=7), who claimed that this is not necessary. Most respondents (n=18; 46%) said that changes have to be tested to the full extent for every planned release. The remaining 36% (n=14) partially agreed that testing is always necessary and specified in the comments specific situations where more testing was necessary or less testing was acceptable. Some stated that the scope of testing depends on the number of changes made or on the development phase. One respondent commented that it is not possible to test all changes; another one said that full testing is always necessary because cross-dependencies only become visible by testing within a release.

Only 10% (n=4) of the respondents agreed that all types of tests have to be performed in every release cycle (Q30). 39% (n=15) disagreed with this statement and about half of them (n=20; 51%) partially agreed. The participants further pointed out that conducting all tests is not feasible or that the necessary types of tests are predefined in the test strategy and depend on the change itself. Others reported that regression tests are often sufficient, or that full releases need to be tested more accurately than partial releases.

To save testing effort, it is important to know when comprehensive testing (including all types of tests) of all ECUs needs to be done (Q31). 85% of the respondents (n=33) answered that testing needs to be done dependent on the software, hardware, or functional changes. Five respondents (13%) claimed that the ECUs have to be tested once per qualification phase, either at the beginning or at the end. 2% (n=1) said that testing is not always necessary. One participant commented that due to the high product complexity and low software quality, all ECUs have to be tested as an integrated system with all possible tests, or at least with good regression tests. Another one claimed that comprehensive testing is not possible for all system parts, but major parts can be covered with a good testing strategy.

C.RQ2: AGILE-SPECIFIC CHALLENGES

Existing vehicle development processes emerged at a time when agility was not present yet and were formalized based on traditional development methodologies. Due to the regulations, strict production deadlines and the complexity in vehicle development, the need to have formal processes will remain. However, the potential to integrate agile processes must be evaluated in order to exploit the benefits of agility. New technologies such as cloud services implicate a stronger customer focus, to be able to respond more flexibly to customer needs, which results in conflicts with the slow and unresponsive traditional development. Innovation is happening fast in the automotive company, and companies have to react in time to stay competitive.

Iterative cycles are already incorporated into many processes, but other concepts of agile methods initially designed for small teams are more difficult to integrate or synchronize with the existing rigid processes. The OEMs are currently performing a balancing act between fixed framework conditions and scope for flexibility. On the one hand, legal requirements, standards and production requirements must be observed and on the other hand, developers want to act more freely without being restricted by guidelines. The results of this survey indicate that this is not a simple procedure.

The survey revealed that if departments are already working with agile methods, they only use them to a certain extent. Our initial expectation was that agile methods are commonly used at least in fields such as connected car, with its digital services and shorter development cycles. The differences between our expectations and reality may be caused by the lack of a common understanding of agile methods. This is confirmed by the inconsistency of the answers by the respondents, who considered additional qualification phases necessary but at the same time did not demand more qualification phases. The reason for this may be a lack of knowledge about agile methods.

There is also a lack of suitable means of communication for short, regular exchanges aimed at establishing transparency between all participants. Such possibilities for fast feedback would also increase the overall quality of voting and benefit the flow of information. Respondents

(n=7) from agile projects reported that the length of release cycles is too long and does not suit agile approaches.

The fact that management decisions have such a strong influence on the further course of development illustrates that decisions are made at higher levels of hierarchy. In an optimal agile environment, the development team makes the decisions. Based on the priorities set by the Product Owner and the requirements dependencies identified by the development team, a Scrum team knows best how to achieve the best solutions. At the beginning of each iteration, they commit to a product increment that is valuable and achievable. If management forces decisions upon the team during an iteration, results can be expected to be suboptimal.

However, this is only the point of view of a single team. If each single team cannot meet their commitments, the qualification phase of an integrated product is going to raise problems. One reason is that the release plan, which considers dependencies between different projects, gets unofficially changed without being updated. That means the developers change their release plans on the operational level without having the change approved and without informing the affected interfaces.

D.RQ3. IMPROVEMENTS WITH AGILE METHODS

There are many challenges that are predestinated to be solved with agility. The survey revealed that transparency and coordination are highly important for a successful qualification phase. Some of the interviewees states that the communication path in their department is too long, which causes loss of time and a lack of coordination. This argument is supported by the fact that some of the participants reported not knowing their interfaces and relevant stakeholders, resulting in bad synchronization and integration structures. By using agile development and small working groups with no typical hierarchy, interface management and short communication paths could become possible [26].

Currently, additional qualification phases are started to fix the remaining bugs or to finish some functionalities that had been planned for the previous release cycle. Due to the increased effort for these activities, the planned results for the next release cycle cannot be fully achieved, pushing a wave of additional efforts, e.g., for coordinating additional qualification phases, through the whole project. Increased transparency regarding the content that was finished in an iteration can be achieved with a definition of done and by incorporating time-boxed sprints. At the end of each sprint, the status quo is assessed, and unfinished requirements can be planned for the next sprint.

Another characteristic of sprints is that requirements are usually not changed, especially not from outside the team. This would also help to stabilize the release plan, which would help to achieve higher-quality products delivered for integration by each single team. Sprints are usually short iterations of several weeks. Respondents from agile projects reported that the length of release cycles is too long, and that they would prefer receiving feedback earlier. This issue leads to work overload and defined timelines not being

achieved, which ultimately leads to lower software quality. In addition, development costs increase due to many additional qualification phases. By using agile methods and more intermediate steps, including regular assessments of the project state, discrepancies could be identified earlier.

Agile teams use face-to-face communication and daily standups to synchronize their work in order to achieve their sprint goal. In a scaled environment, so-called Scrum-of-Scrums are scaled daily standups where representatives of different teams synchronize their development status and plan their dependencies. The Scaled Agile Framework (SAFe) uses an architectural runway to coordinate architectural decisions between the single development teams to facilitate integration.

Continuous integration is commonly used in agile projects and could be of benefit in qualification phases. Integrating smaller work products incrementally can replace a larger and more complex final integration and provides early transparency about the finished content of the release as well as raising awareness of dependencies.

In general, regular retrospectives can be held at the end of each sprint, helping the team to raise issues impeding their work and improve their development process. Conducting retrospectives together with relevant stakeholders and dependent projects helps to continuously improve collaboration between teams.

V. CONCLUSION AND FUTURE WORK

Agile development is being increasingly adopted even in regulated domains such as the automotive domain. There it has to be synchronized with co-existing traditional development approaches. Qualification phases of traditional, hybrid, and agile projects are difficult. An online survey was used to identify challenges in the release planning of a German automotive OEM. The results show that communication and information issues such as inadequate familiar counterparts predominate in the case company. Dependencies between input and output relations are key results, too. Furthermore dissatisfied software quality during the system tests leads to further challenges. Another key statement of the survey results is the limited capacity to act to the supplier relationships.

We presented the main challenges in detail and discussed the state of agility and the conflicts arising in the context of co-existing traditional and agile approaches. We outlined how agile concepts could improve some of the identified challenges and thus provided recommendations for practitioners.

In the future, we plan an in-depth analysis and interpretation of the results, including a more detailed analysis of the questions, by trying to identify further correlations. With a questionnaire adapted to collect experiences outside the case company, we want to check whether there are similar problems at other automotive companies or even companies from other regulated domains that are developing complex systems in a hybrid project environment.

REFERENCES

- [1] VersionOne: The 13th annual state of agile report. (2019). www.col-lab.net
- [2] Hohl, P., Münch, J., Schneider, K., Stupperich, M. (2016). Forces that prevent agile adoption in the automotive domain. In *proc. of the International Conference on Product-Focused Software Process Improvement, PROFES 2016: Product-Focused Software Process Improvement* (pp 468-476). DOI.org/10.1007/978-3-319-49094-6_32.
- [3] Diebold, P., Zehler, T. (2016). The right degree of agility in rich processes. In *Managing Software Process Evolution* (pp. 15-37). DOI: 10.1007/978-3-319-31545-4_2.
- [4] Diebold, P., Theobald, S. (2018). How is agile development currently being used in regulated embedded domains? In *proc. of JSEP'18*. DOI.org/10.1002/smr.1935.
- [5] Klünder, J., Hohl, P., Fazal-Baqaie, M., Krusche, S., Küpper, S., Linssen, O., & Prause, C.R. (2017). HELENA study: reasons for combining agile and traditional software development approaches in german companies. In *proc. of PROFES'17*. DOI.org/10.1007/978-3-319-69926-4_32.
- [6] Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektere, K., McCaffery, F., Linssen, O., Hanser, E., Prause, C. R. (2017). Hybrid software and system development in practice: waterfall, scrum, and beyond. In *proc. of the 2017 International Conference on Software and System Process* (pp. 30-39). <https://doi.org/10.1145/3084100.3084104>.
- [7] Theobald S., Diebold P. (2018): Interface Problems of Agile in a Non-agile Environment. In: Garbajosa J., Wang X., Aguiar A. (eds) *Agile Processes in Software Engineering and Extreme Programming, XP 2018. Lecture Notes in Business Information Processing*, vol 314. Springer, Cham. DOI.org/10.1007/978-3-319-91602-6_8.
- [8] Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S. G., Münch, J., Pfahl, D., Linssen, O., Kuhrmann, M. (2019). What are Hybrid Development Methods Made Of? An Evidence-based Characterization. In *proc. of the ICSSP19*. DOI.org/10.1109/ICSSP.2019.00022.
- [9] Broy, M. (2006). Challenges in automotive software engineering. In *proc. of the 28th international conference on Software engineering* (pp. 33-42). ACM. DOI.org/10.1145/1134285.1134292
- [10] Hohl, P. (2019). An assessment model to foster the adoption of agile software product lines in the automotive domain. University of Hannover, Hannover, Germany. DOI.org/10.1109/ICE.2018.8436325.
- [11] Saliu, O., Ruhe, G. (2005). Supporting Software Release Planning Decisions for Evolving Systems. In *proc. of NASE SEW-29*, Washington DC. DOI: 10.1109/SEW.2005.42.
- [12] Heikkilä, V. T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T., & Järvinen, J. (2015). Operational release planning in large-scale scrum with multiple stakeholders—A longitudinal case study at F-Secure Corporation. In *Information and Software Technology*, 57, (pp.116-140). DOI.org/10.1016/j.infsof.2014.09.005.
- [13] Sax, E., Reussner, R., Guissouma, H., Klare, H. (2017). A Survey on the state and future of automotive software release and configuration management. *Karlsruhe Reports in Informatics*, 11. DOI: 10.5445/IR/1000075673.
- [14] Bestfleisch, U., Herbst, J., & Reichert, M. (2005). Requirements for the workflow-based support of release management processes in the automotive sector. In *proc. of the 12th European Concurrent Engineering Conference ECEC'05*.
- [15] Müller, D., Herbst, J., Hammori, M., Reichert, M. (2006). IT support for release management processes in the automotive industry. In *proc. of the International Conference on Business Process Management* (pp. 368-377).
- [16] Lindgren, M., Land, R., Norström, C., Wall, A. (2008). Key aspects of software release planning in industry. In *proc. of the 19th Australian Conference on Software Engineering* (pp. 320-329). DOI: 10.1109/ASWEC.2008.4483220.
- [17] Danesh, A. S., Ahmad, R. B., Saybani, M. R., & Tahir, A. (2012). Companies approaches in software release planning-based on multiple case studies. *JSW*, 7(2), 471-478. DOI:10.4304/jsw.7.2.471-478.
- [18] Heikkilä, V. T., Paasivaara, M., Lassenius, C., & Engblom, C. (2013). Continuous release planning in a large-scale scrum development organization at ericsson. In *proc. of the International Conference on Agile Software Development* (pp. 195-209). DOI.org/10.1007/978-3-642-38314-4_14.
- [19] Heikkilä, V., Rautiainen, K., & Jansen, S. (2010). A revelatory case study on scaling agile release planning. In *proc. of the 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications* (pp. 289-296). DOI: 10.1109/SEAA.2010.37.
- [20] Karvonen, T., Behutiye, W., Oivo, M., & Kuvaja, P. (2017). Systematic literature review on the impacts of agile release engineering practices. *Information and Software Technology*, 86, (pp. 87-100). <https://doi.org/10.1016/j.infsof.2017.01.009>.
- [21] Ameller, D., Farré, C., Franch, X., & Rufian, G. (2016). A survey on software release planning models. In *International Conference on Product-Focused Software Process Improvement* (pp. 48-65). DOI: 10.1007/978-3-319-49094-6_4.
- [22] M. Svahnberg, T. Gorschek, R. Feldt, R. Torkar, S. B. Saleem, and M. U. Shafique (2010). A systematic review on strategic release planning models. *Information and Software Technology* (pp. 237–248). DOI: 10.1016/j.infsof.2009.11.006
- [23] Carlshamre, P. (2002). Release planning in market-driven software-product development: Provoking an understanding. *Requir. Eng.* 7(3), (pp. 139–151). DOI: 10.1007/s007660200010.
- [24] Jantunen, S., Lehtola, L., Gause, D. C., Dumdum, U. R., Barnes, R. J. (2011). The challenge of release planning. In *proc. of the Fifth International Workshop on Software Product Management*, (pp. 36–45). DOI: 10.1109/IWSPM.2011.6046202.
- [25] Benestad, H.C., Hannay, J.E (2011). A comparison of model-based and judgment-based release planning in incremental software projects. In *proc. of the 33rd International Conference on Software Engineering*, (pp. 766–775). DOI: 10.1145/1985793.1985901.
- [26] Spiegler S.V., Heinecke C., Wagner S. (2019). Leadership Gap in Agile Teams: How Teams and Scrum Masters Mature. In: Kruchten P., Fraser S., Coallier F. (eds) *Agile Processes in Software Engineering and Extreme Programming, XP 2019. Lecture Notes in Business Information Processing*, vol 355. Springer, Cham. (pp 37-52) DOI: 10.1007/978-3-030-19034-7_3